

Aplikativni protokoli

Aplikativni sloj je sedmi, odnosno najviši sloj OSI modela. Sva direktna manipulacija podataka i informacija se odvija na ovom nivou i time se omogućava pristup softveru i korisniku. Neki od servisa koje ovaj sloj pruža su: elektronska pošta, slanje i prijem fajlova, mrežni resursi, distribucija podataka korisniku itd.

Ovaj sloj se sastoji od različitih vrsta protokola. Jedan od njih je široko rasprostranjeni HTTP (HyperText Transfer Protocol), koji je osnova svetske računarske mreže (World Wide Web). Kad god u našem pretraživaču otvaramo stranu, pretraživač serveru, koji koristi HTTP, šalje ime željene strane i njenu lokaciju i server je šalje nazad. U ovoj nastavnoj jedinici ćemo se, pored upoznavanja sa aplikativnim protokolima, kroz praktičan rad upoznati i sa FTP-om. Koristeći ftplib Python biblioteku, vršićemo izlistavanje foldera i fajlova sa eksternog FTP servera, kretanje kroz strukturu i preuzimanje i otpremanje fajlova.

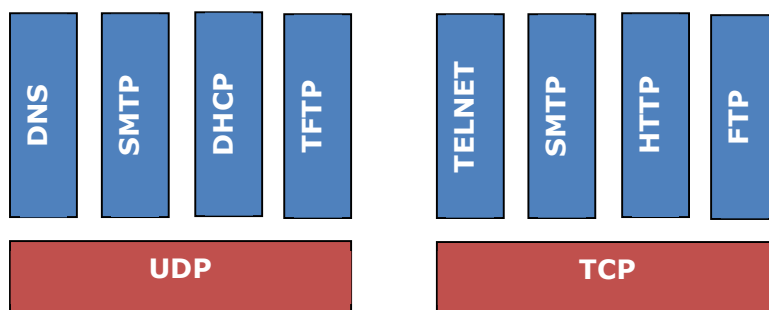
Napomena:

World Wide Web i HTTP su usko povezani. Naime, World Wide Web predstavlja informacioni sistem gde se dokumenta i ostali resursi adresiraju pomoću URL-ova (Uniform Resource Locators), koji se dobavljaju i šalju preko HTTP-a. Dakle, World Wide Web nije sam internet, već samo njegov deo, iako se ta dva pojma često poistovećuju.

Drugi aplikativni protokoli su:

- FTP (File Transfer Protocol) – omogućava korisniku slanje i povlačenje fajlova sa udaljene mašine. Koristi TCP protokol. Prilikom povezivanja sa udaljenim uređajem ostvaruju se dve konekcije i to: data konekcija i kontrolna konekcija. Data konekcija koristi port 20, a kontrolna konekcija port 21. Jedna konekcija (data) se koristi isključivo za transfer podataka, dok druga (kontrolna) služi za slanje komandi i pribavljanje odgovora.
- SMTP (Simple Mail Transfer Protocol) – omogućava korisniku slanje i primanje elektronske pošte. Koristi TCP protokol. SMTP protokol ima dve strane – klijentsku i serversku, i koristi port 25.
- TELNET (remote login) – udaljeni pristup koji omogućava korisniku da sa jedne mašine može slati komande drugoj mašini – posredstvom terminala. Koristi TCP protokol. Ovo čini pomoću NVT (Network Virtual Terminal) sistema, koji kodira poslate komande na klijentskoj strani i dekodira ih na serverskoj strani. Port koji koristi je 23.
- DNS (Domain Name System) – ovaj sistem nam omogućava mapiranje IP adresa sa imenima sajtova na koje te adrese pokazuju. DNS je podeljen na tri kategorije:
 - generički domen – ovaj domen se odnosi na ponašanje sajta – da li je reč o sajtu vlade države (.gov), da li je reč o sajtu organizacije (.org) ili pak edukativne ustanove (.edu); u ovom domenu se koriste tri karaktera;
 - domen zemlje – ovaj domen podrazumeva dvoslovnu skraćenicu zemlje kojoj sajt pripada (.me, .rs, .us); u ovom domenu se koriste dva karaktera;
 - inverzni domen – ovaj domen se koristi da se IP adresa mapira na ime sajta.

Neki od ovih protokola koriste UDP, a neki TCP kao transportni protokol. Podela aplikativnih protokola prema tome koje transportne protokole koriste može se videti na sledećoj slici:



Slika 1.1. Podela aplikativnih protokola prema transportnim protokolima na kojima su bazirani

Funkcije aplikativnog sloja:

- elektronska pošta – ovaj sloj pruža osnovu za slanje, primanje i skladištenje elektronske pošte (e-mail);
- mrežni virtualni terminali – omogućavaju korisniku povezivanje sa udaljenim uređajem; aplikacija emulira terminal na tom udaljenom uređaju, pa se tako ostvaruje komunikacija između korisnika i tog krajnjeg uređaja posredstvom terminala;
- prenos fajlova – korisnik ima mogućnost da pristupi udaljenom fajlu i rukuje njime, a takođe ga može i dobiti;
- procesi na oba kraja (početni i destinacioni uređaji) poslednji su u lancu komunikacije.

Aplikativni protokol definiše kako procesi na datom uređaju (ili dva uređaja) prenose informacije između sebe. Pa tako, protokol aplikativnog sloja definiše:

- tip poruke – da li je reč o porukama kao zahtevima ili porukama kao odgovorima;
- sintaksu poruka – koja su polja u porukama i kako su ta polja u poruci razgraničena;
- značenje polja – značenje i odabir informacije koju će to polje sadržati;
- pravila i uslovi – kako i kada će proces poslati poruke i odgovoriti na poruke.

Pitanje

FTP protokol služi za:

- slanje mejlova
- **slanje fajlova**
- primanje mejlova

Objašnjenje:

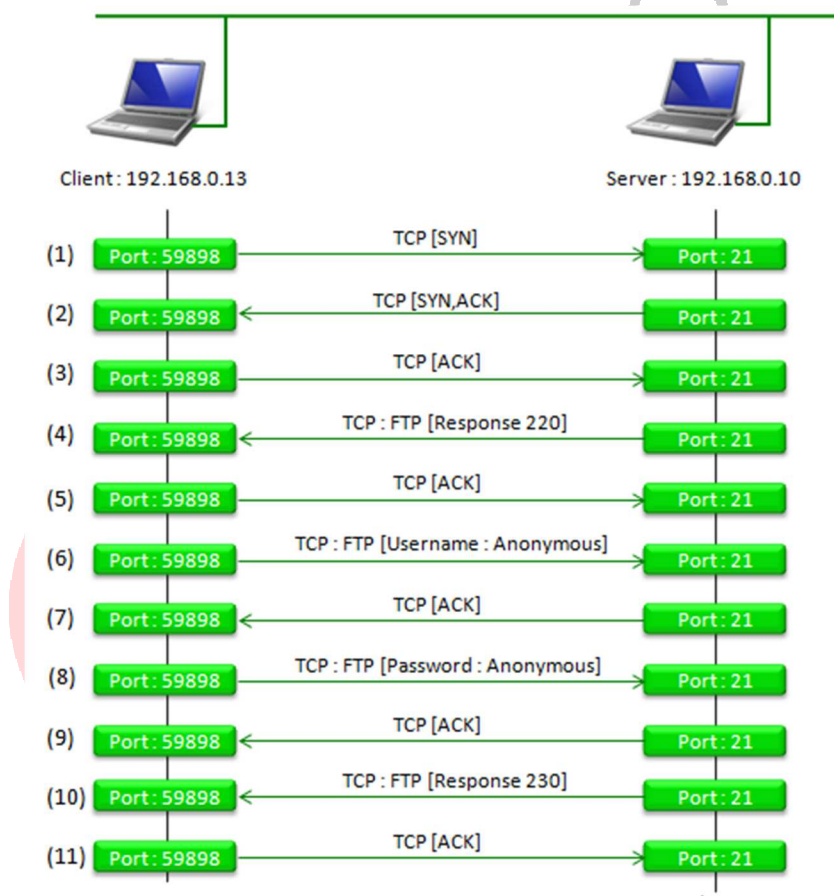
Tačan odgovor je da FTP protokol služi za slanje, isčitavanje, brisanje i listanje fajlova sa date adrese.

FTP protokol

FTP protokol je protokol za transfer fajlova u željeni direktorijum sa jedne mašine na drugu. Osnove ovog protokola datiraju čak iz 1971. godine.

Uobičajena FTP sesija započinje željom jednog korisnika da pošalje fajl drugom korisniku. Za samo slanje fajlova je potrebno da korisnik serveru pošalje svoju identifikaciju i šifru radi autorizacije. Nakon uspješne autorizacije, korisnik može slati fajlove ka serveru i obrnuto.

HTTP i FTP su dosta slični, jer se oba baziraju na sigurnom transferu podataka, a i oba su bazirana na TCP protokolu. Razlike između ova dva protokola se ogledaju u tome što FTP zahteva dve paralelne konekcije za transfer fajlova – kontrolnu konekciju (control connection) i konekciju za podatke (data connection). Kontrolna konekcija se koristi za identifikaciju i autorizaciju klijenta sa serverom, za slanje FTP metoda kojim se omogućava samo slanje ili dobavljanje fajla, za menjanje udaljenog direktorijuma itd. Konekcija za podatke se zapravo koristi za samo slanje fajla, tačnije njegovog sadržaja.



Slika 1.2. Dijagram FTP konekcije¹

¹ https://www.sharetechnote.com/html/IP_Network_FTP.html

Pri započinjanju FTP sesije sa serverom, klijent prvo uspostavlja TCP kontrolnu konekciju na portu 21. Odmah nakon toga klijent šalje svoju identifikaciju i šifru preko kontrolne konekcije. Nakon što je uspešno autorizovan od strane servera, klijent šalje komande za dobavljanje željenog fajla, menjanje udaljenog direktorijuma kako bi došao do željenog fajla, kreiranje novog direktorijuma itd. Nakon što korisnik (klijent) zatraži transfer fajla (bilo da je reč o slanju ka serveru ili od servera), FTP protokol otvara TCP konekciju za prenos fajlova koji se odvija po portu 20. Po jednoj konekciji za prenos fajla se može poslati samo jedan fajl. Ako tokom iste sesije korisnik želi da pošalje ili dopremi još jedan fajl, otvara se nova TCP konekcija za prenos podataka, pa tako kontrolna konekcija ostaje otvorena kroz čitavu sesiju. Takođe, za čitavu sesiju server mora voditi računa o tome kom klijentu je dodeljena koja kontrolna konekcija.

FTP metode

FTP metode (komande) koje se šalju serveru se sastoje od najčešće četiri slova, gde je ime svake od tih metoda sedam bita u ASCII formatu. Neke od tih metoda su:

METODA	OPIS
USER	metoda kojom šaljemo ime korisnika koji želi da se poveže na server i inicira FTP konekciju
PASS	metoda kojom šaljemo šifru korisnika koji želi da se poveže na server i inicira FTP konekciju
LOGIN	metoda kojom klijent podnosi informacije ka serveru (slanje log-in forme)
CWD	metoda kojom se omogućava korisniku da promeni udaljeni direktorijum
RMD	metoda kojom se omogućava korisniku da obriše željeni direktorijum
RETR	metoda kojom se omogućava korisniku da inicira konekciju za prenos podataka kako bi dobio (ili poslao) fajl
STOR	metoda kojom se omogućava korisniku da uskladišti fajl u željeni direktorijum
LIST	metoda koja omogućava korisniku da izlista fajlove u trenutnom direktorijumu
ABOR	metoda kojom klijent nagoveštava serveru da prekine trenutnu konekciju za slanje podataka
QUIT	metoda kojom klijent okončava kontrolnu konekciju sa serverom

Tabela 1.1. Tabelarni prikaz FTP metoda

FTP statusni kodovi i poruke

Svaki od zahteva ka serveru šalje određeni kod kome odgovara određena statusna poruka. Neki od tih kodova i poruka koje im pripadaju su:

Statusni kod	Statusna poruka
200	Command okay
221	Service closing control connection
225	Data connection open; no transfer in progress
331	User name okay, need a password
502	Command not implemented
503	Bad sequence of commands

504	Command not implemented for that parameter
530	Not logged in
551	Requested action aborted: page type unknown

Tabela 1.2. Tabela prikaz FTP statusnih kodova i poruka

Anonimni FTP (Anonymous FTP)

Anonimni FTP je podvrsta FTP protokola u kojoj nije potrebna autorizacija klijenta jer se korisničko ime i šifra postavljaju na podrazumevani anonymous. Anonimni FTP se koristi uglavnom za direktorijume i fajlove koji su javno dostupni i u javnom domenu.

Čitanje FTP direktorijuma pomoću Pythona

Povezivanje na FTP server u Pythonu se vrši koristeći ugrađenu ftplib biblioteku. Radi jednostavnosti, koristićemo javno dostupan server (anonimni FTP) speedtest4.tele2.net, čije je korisničko ime za logovanje anonymous, dok se šifra prosleđuje kao prazno polje. Povezivanje na taj server se vrši jednostavnim korišćenjem FTP klase ove biblioteke:

Rad sa FTP serverom (ftpexample.py):

```
from ftplib import FTP
ftp = FTP('speedtest4.tele2.net')
ftp.login('anonymous','')
ftp.dir()
ftp.cwd('upload')
print('Content of folder: upload')
ftp.dir()
ftp.close()
```

```
C:\WINDOWS\system32\cmd.exe
> python ftpexample.py
-rw-r--r-- 1 0 0 1073741824000 Feb 19 2016 100GB.zip
-rw-r--r-- 1 0 0 107374182400 Feb 19 2016 100GB.zip
-rw-r--r-- 1 0 0 102400 Feb 19 2016 100KB.zip
-rw-r--r-- 1 0 0 104857600 Feb 19 2016 100MB.zip
-rw-r--r-- 1 0 0 10737418240 Feb 19 2016 10GB.zip
-rw-r--r-- 1 0 0 10485760 Feb 19 2016 10MB.zip
-rw-r--r-- 1 0 0 1073741824 Feb 19 2016 1GB.zip
-rw-r--r-- 1 0 0 1024 Feb 19 2016 1KB.zip
-rw-r--r-- 1 0 0 1048576 Feb 19 2016 1MB.zip
-rw-r--r-- 1 0 0 209715200 Feb 19 2016 200MB.zip
-rw-r--r-- 1 0 0 20971520 Feb 19 2016 20MB.zip
-rw-r--r-- 1 0 0 2097152 Feb 19 2016 2MB.zip
-rw-r--r-- 1 0 0 3145728 Feb 19 2016 3MB.zip
-rw-r--r-- 1 0 0 524288000 Feb 19 2016 500MB.zip
-rw-r--r-- 1 0 0 52428800 Feb 19 2016 50MB.zip
-rw-r--r-- 1 0 0 524288 Feb 19 2016 512KB.zip
-rw-r--r-- 1 0 0 5242880 Feb 19 2016 5MB.zip
drwxr-xr-x 2 105 108 77824 Aug 24 01:51 upload
Content of folder> upload
-rw----- 1 105 108 48580 Aug 24 01:49 34q6df7s88ph0ap55kjmric5q.txt
-rw----- 1 105 108 80088064 Aug 24 01:51 Copy VID_20200813_001221.mp4
-rw----- 1 105 108 69235771 Aug 24 01:51 MIDE-134.mp4
-rw----- 1 105 108 2852685061 Aug 24 01:51 MIFD-115.mp4
-rw----- 1 105 108 3643453653 Aug 24 01:51 MIRD-203.mp4
-rw----- 1 105 108 3481304526 Aug 24 01:51 MMYM-034.mp4
-rw----- 1 105 108 3706919105 Aug 24 01:51 MOPG-059.mp4
-rw----- 1 105 108 3525317763 Aug 24 01:51 MVSD-414.mp4
-rw----- 1 105 108 3766610096 Aug 24 01:51 NACR-293.mp4
-rw----- 1 105 108 4588981363 Aug 24 01:51 NGOD-118.mp4
-rw----- 1 105 108 4033313147 Aug 24 01:51 NGOD-125.mp4
-rw----- 1 105 108 4940336076 Aug 24 01:51 NITR-476.mp4
-rw----- 1 105 108 1000000 Aug 24 01:49 cscscsf5fttbapassspeeu9j.txt
-rw----- 1 105 108 716208 Aug 24 01:51 qcb2f40d8tr97ujc82gpgf26ua.txt
```

Slika 1.3. Ispis liste fajlova korenskog direktorijuma i direktorijuma upload

Kreiranjem instance FTP klase smo omogućili povezivanje na speedtest4.tele2.net koristeći anonymous kao korisničko ime preko login() metode ove klase. U slučaju da smo prosledili pogrešne kredencijale, ova biblioteka bi nam ispislala getaddrinfo grešku. Nakon uspešnog povezivanja možemo, u domenu ovlašćenja naloga sa kojim smo se ulogovali (ime naloga u ovom slučaju je anonymous), čitati i preuzimati fajlove i listati foldere. Menjanje postojećih fajlova na serveru ili slanje novih fajlova na server nije dozvoljeno kada se koristi ovaj server. U ispisanim redovima metode dir() vidimo dva različita ispisa: jedan koji se odnosi na fajlove (npr: -rw-r--r-- 1 0 0 524288 Feb 19 2016 512KB.zip, -rw-r--r-- 1 0 0 1048576 Feb 19 2016 1MB.zip) i drugi koji se odnosi na direktorijume: drwxr-xr-x 2 105 108 77824 Aug 24 01:51 upload (konkretno se ova linija odnosi na folder upload). Ako bismo želeli da se pozicioniramo i izlistamo upravo taj upload folder, koristimo metodu cwd('upload'), kojoj prosleđujemo putanju do foldera kojem želimo da pristupimo, koja odgovara jednoj od metoda FTP protokola o kojima je bilo reči ranije u lekciji. Nakon završetka rada sa FTP klasom, dobra je praksa i ručno zatvoriti konekciju sa FTP serverom naredbom ftp.close().

Za dodatne informacije o dostupnim metodama pogledati [zvaničnu dokumentaciju](#).

Dobavljanje fajla preko FTP protokola (download)

Kroz FTP biblioteku možemo, kao što i specifikacija FTP protokola nalaže, i preuzimati fajlove. Za preuzimanje fajlova koristimo naredbu FTP protokola RETR. Ovu naredbu koristimo zajedno sa metodom retrbinary FTP sesijskog objekta, kojoj prosleđujemo string u formatu

RETR ime_fajla_koji_preuzimamo, kao i poziv metode write nad fajl objektom, koja nam je potrebna radi smeštanja sadržaja fajla sa servera u lokalni fajl:

ftpexample.py fajl za dobavljanje fajla sa servera:

```
from ftplib import FTP
ftp = FTP('speedtest4.tele2.net')
ftp.login('anonymous','')
local_filename = open('1KB.zip', "wb")
ftp.retrbinary("RETR {}".format('1KB.zip'), local_filename.write)
local_filename.close()
ftp.close()
```

Iz prethodnog primera (primer o listanju direktorijuma) uvideli smo koji su nam fajlovi dostupni u korenskom direktorijumu servera. Iz te liste možemo odabrati bilo koji fajl za dobavljanje. Izabraćemo fajl po imenu 1KB.zip.

Ovaj primer funkcioniše tako što prvo otvaramo lokalni fajl (fajl na našem disku) u koji želimo da smestimo sadržaj sa servera. Nazvali smo ga isto kao i fajl koji želimo da dobavimo – 1KB.zip. Otvaramo ga u binarnom režimu – wb. Sada koristimo retrbinary metodu FTP sesijskog objekta kojem prosleđujemo string RETR 1KB.zip, kao i write metodu otvorenog fajla. Ovaj način, prosleđivanja funkcije kao parametra, naziva se callback funkcija. Po izvršavanju linije ftp.retrbinary(..), fajl je sada smešten na našem disku i možemo sigurno zatvoriti i naš local_filename otvoreni fajl.

Važno je napomenuti da su kod ovog konkretnog FTP servera – <ftp://speedtest4.tele2.net/> – smešteni fajlovi prazni i da će svaki pokušaj njihovog otvaranja i čitanja nakon što smo ih dobavili rezultirati greškom. Ovo uglavnom nije slučaj i fajlovi preuzeti na ovakav način se mogu otvoriti. Ovo važi samo za ovaj server.

Otpremanje fajla na FTP server (upload)

Otpremanje (upload) fajla vršimo po sličnom principu kao i dobavljanje. Komanda FTP protokola za upload fajla je STOR, pa je tako metoda FTP sesijskog objekta koju ćemo koristiti storbinary; prosleđujemo joj string oblika STOR ime_fajla_koji_otpremamo, kao i otvoreni fajl objekat:

ftpexample.py fajl za otpremanje fajla sa servera:

```
from ftplib import FTP
ftp = FTP('speedtest4.tele2.net')
ftp.login('anonymous','')
ftp.cwd('upload')
to_upload = open('ftpexample.py','rb') # file to send
ftp.storbinary('STOR ftpexample.py', to_upload) # send the file
to_upload.close()
ftp.close()
```

Konfiguracija <ftp://speedtest4.tele2.net/> servera je takva da se fajlovi mogu otpremati samo u folder upload koji se nalazi u korenskom direktorijumu servera. Nakon što su fajlovi otpremljeni, odmah se i brišu, što se može pročitati u zvaničnom uputstvu ovog [servisa](#).

Zato, moramo se prvo pozicionirati u folder upload naredbom `ftp.cwd('upload')`. Nakon toga otvorimo za čitanje fajl koji želimo da pošaljemo (prvo ga moramo iščitati). Fajl koji smo ovde otvorili je zapravo i naš skript po imenu `ftpexample.py`. Za slanje tog fajla u direktorijum upload koristimo metodu `storbinary`, kojoj prosleđujemo string koji predstavlja FTP komandu: `STOR ftpexample.py`, kao i objekat otvorenog fajla `to_upload`. Izvršenjem ove linije smo uspešno postavili fajl na server. Pošto konfiguracija servera tako nalaže, ovaj fajl će se obrisati, ali će se u realnim uslovima fajl zadržati na serveru gde je i postavljen. Ako otpremanje fajla nije uspešno, dobićemo grešku, pa tako, ako se, na primer, nismo pozicionirali u jedini folder na serveru gde je moguće otpremanje korisničkih fajlova – folder upload, dobićemo grešku `ftplib.error_perm: 553 Could not create file`, koja nam govori da nemamo adekvatne dozvole za otpremanje. Više o statusnim kodovima i greškama se može naći na ovom [linku](#).

Rezime

- Aplikativni sloj je najviši sloj OSI modela.
- Najkorišćeniji protokol aplikativnog sloja je HTTP, koji je baziran na TCP transportnom protokolu.
- Aplikativni protokol definiše tip poruke, sintaksu poruka, značenje polja, pravila i uslove prilikom komunikacije.
- FTP protokol je zadužen za slanje i primanje fajlova na mreži.
- FTP protokol je baziran na TCP protokolu.
- Za slanje fajlova preko FTP-a se kreiraju dve konekcije: kontrolna konekcija i konekcija za prenos samog fajla.
- Kontrolnom konekcijom autorizujemo klijenta kako bismo mu omogućili FTP konekciju.
- Kontrolna konekcija se odvija po portu 20, a konekcija za prenos podataka po portu 21.
- Anonimni FTP se koristi uglavnom za direktorijume i fajlove koji su javno dostupni i u javnom domenu.